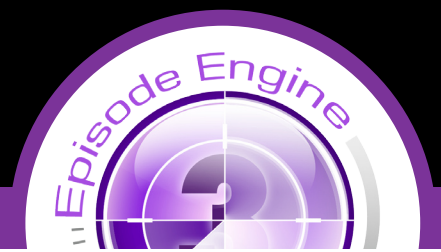# Episode Engine

Best Practices - Deployment

Episode® Engine is a server-based encoder providing extensive format support and superior quality in combination with top performance. The powerful product, at a very affordable price point, makes Episode Engine the obvious choice for your media production workflow. This document will guide you through a few examples of best practice deployments for various organizational needs, from a single Episode Engine setup for smaller production facilities to larger cluster deployments for high volume, time-critical workflows.
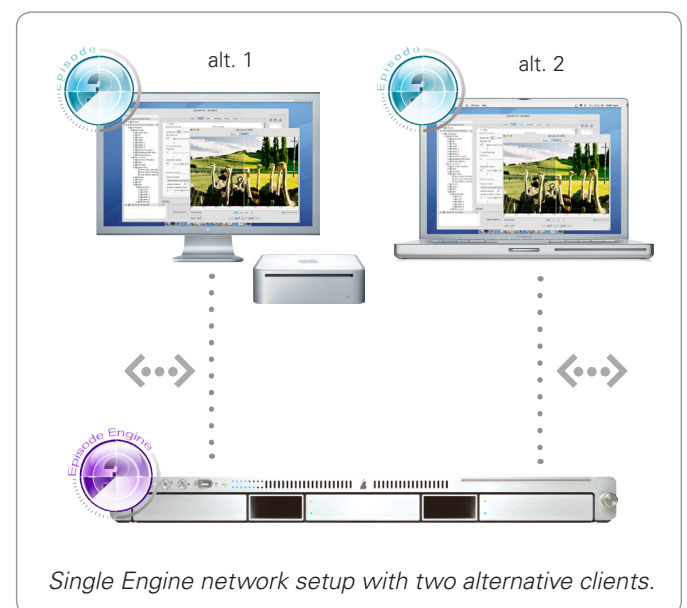
Single Engine Deployment

The Single Engine deployment is recommended for smaller productions with requirements on an automated workflow, but with a moderate amount of files to encode per day. The deployment consists of an Episode running on a Mac Mini and Episode Engine running on an Xserve or Episode and Episode Engine running on a Mac Pro. Note: For more information on single Engine hardware setup, see the Hardware Setup document.

Episode can be used for desktop transcoding and for creating settings for Episode Engine. Over a hundred preset and editable transcoding settings templates let you get started immediately with very high-quality transcoding. With the extensive options for fine-tuning of filters and media format, Episode ensures maximum output quality.

Create your settings in Episode and upload it to Episode Engine. Use Engine Admin to setup an input monitor choosing your exported setting or one of the hundreds of predefined templates. As soon as the input monitor recieves a file, Episode Engine will automatically process it according to the rules in the setting. This processing is called a job. Episode Engine can handle several jobs that are processed in parallel.
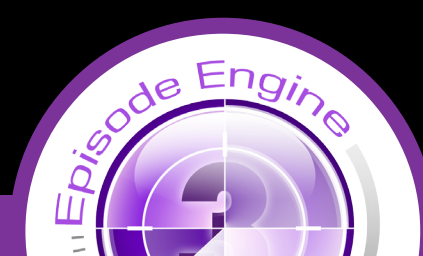

Single Engine setup on an eight core Mac Pro.

Single Engine network setup with two alternative clients.

Flip4Mac®
Digital Media Tools for the Mac

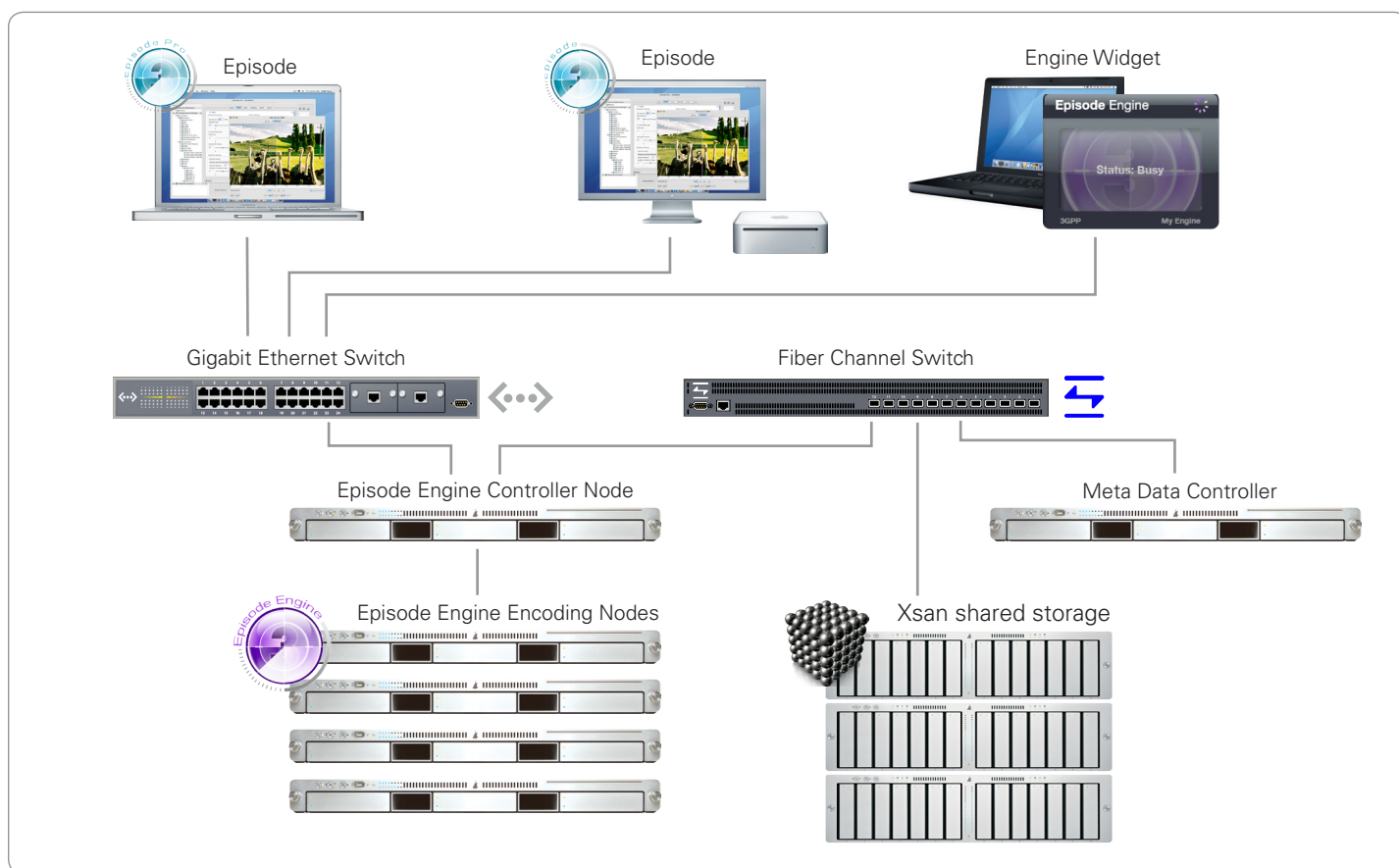**TELESTREAM**

Episode Engine in Larger Deployments

Telestream's Episode Engine software is all about high-speed and high-quality compression. When you need to increase your throughput or shorten your processing time even further, add more servers and create an Episode Engine compression cluster.

A cluster setup consists of a controller node and multiple encoding nodes. The controller node manages jobs and distributes them to the different encoding nodes. The jobs are processed in parallel, the throughput is maximized by keeping all the nodes working. As more encoding nodes are added you get a close-to-linear speed improvement.

Deploying clusters with external storage

Episode Engine is designed for continuous scale-up. If a single transcoding node is not sufficient for your needs, you can build a transcoding cluster. A cluster can consist of any number of encoder nodes connected to shared storage. A controller node monitors the encoders and distributes jobs to them, balancing the load so that each CPU is used to its maximum capacity.

For smaller clusters, sharing files with NFS from the controller node may be sufficient, but as you add nodes to your cluster you will need more powerful and faster storage. An Xsan installation gives increased dependability and disk bandwidth for a cluster beyond a few nodes. If your needs increase so much that disk and/or network bandwidth becomes a bottleneck you can set up several clusters in parallel with separate storage and local networks. You can still monitor them all from a single workstation with Engine Admin and generate settings with Episode from wherever you want.

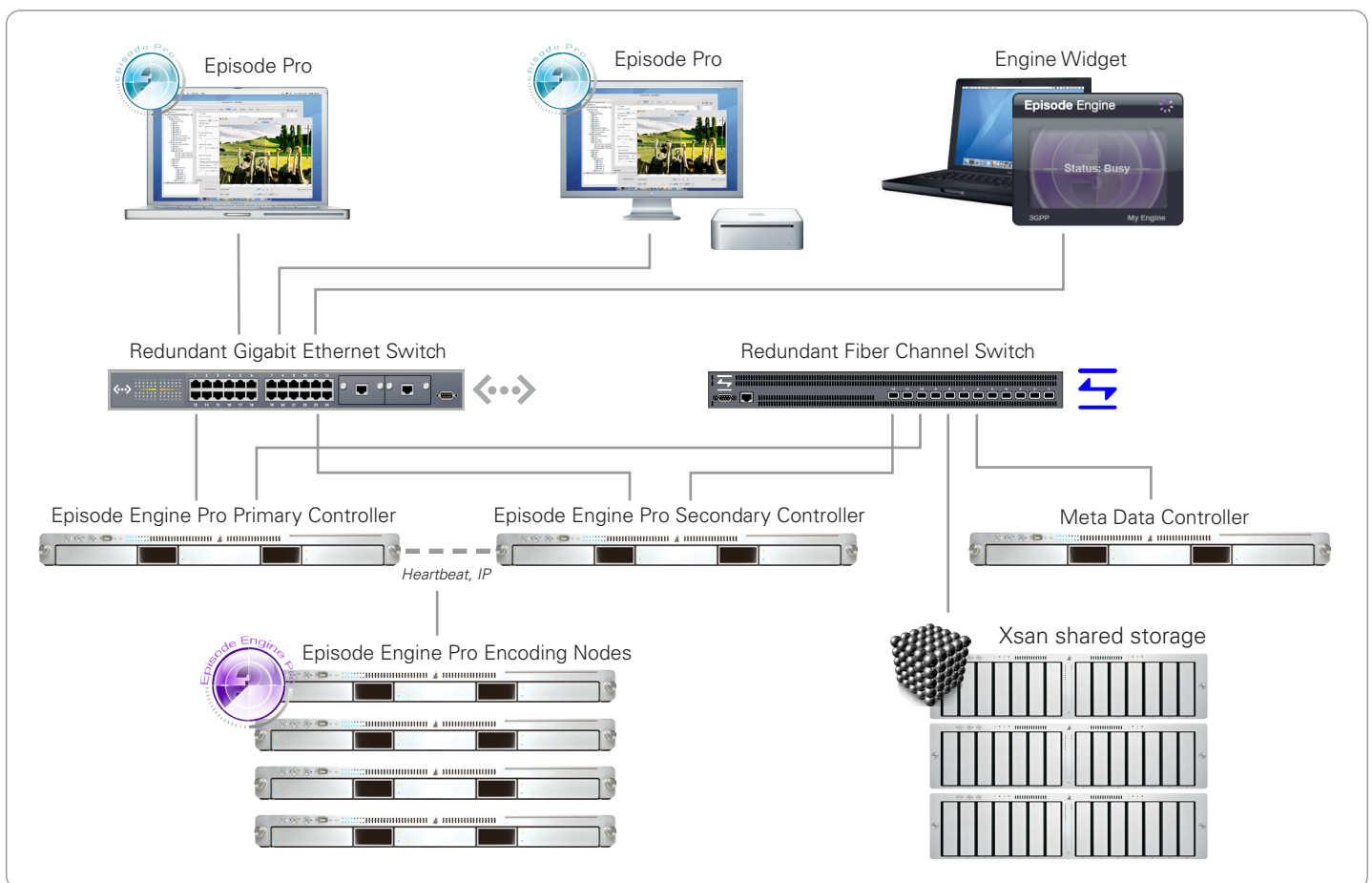Episode   Episode   Engine Widget

Gigabit Ethernet Switch   Fiber Channel Switch

Episode Engine Controller Node   Meta Data Controller

Episode Engine Encoding Nodes   Xsan shared storage

Episode Engine

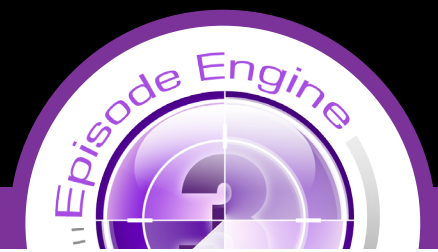## Best Practices - Deployment

Failover

Running server-based Episode Engine software on Mac OS X is the natural choice for maximum robustness. Mac OS X is based on the proven BSD UNIX version with the efficient Mac kernel, and is the result of over 30 years' experience in developing a stable and reliable operating system.

For critical workflows requiring failsafe, non-interrupted service, the Episode Engine Pro solution offers a Failover option. The Failover solution uses a secondary controller node that takes over instantly if the primary controller node fails. The two controller nodes constantly run diagnostics on each other and on the network, and they will alert the operator if any problems are detected. No jobs or information will be lost in the case of failure.

Episode Pro

Episode Pro

Engine Widget

Redundant Gigabit Ethernet Switch

Redundant Fiber Channel Switch

Episode Engine Pro Primary Controller

Episode Engine Pro Secondary Controller

Meta Data Controller

Heartbeat, IP

Episode Engine Pro Encoding Nodes

Xsan shared storage

Flip4Mac®
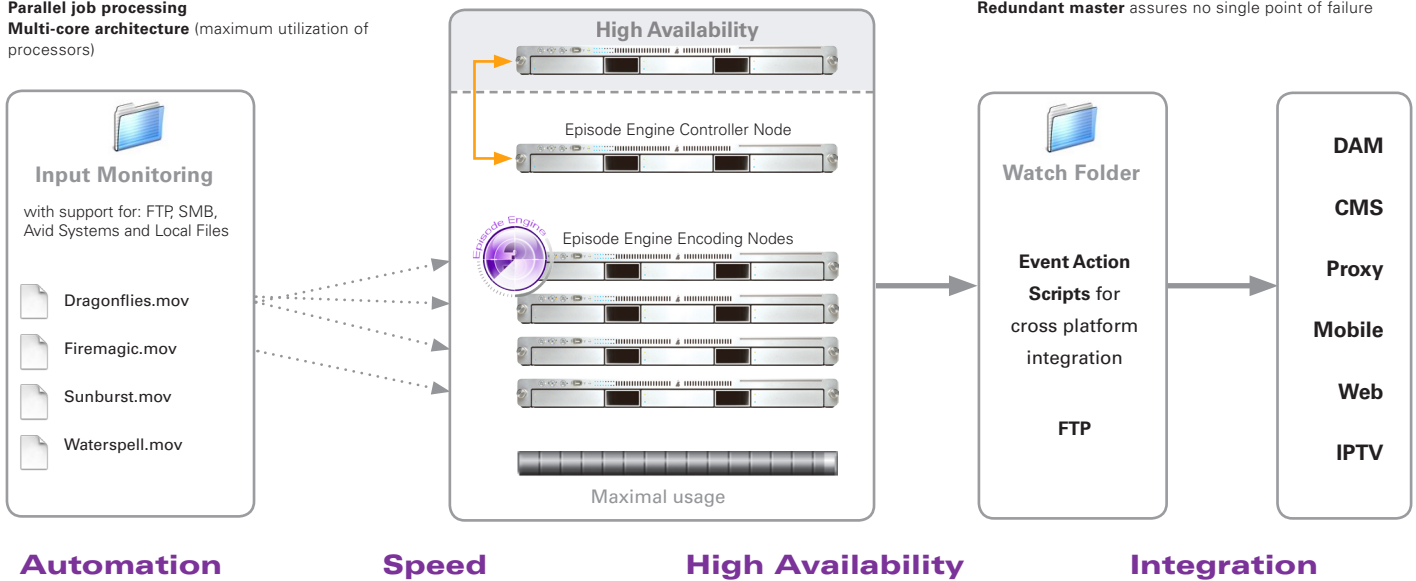Digital Media Tools for the Mac

TELESTREAM

Split & Stitch

If the length of your video material is the bottleneck, rather than the number of required different output files, you can use Episode Engine Pro with the Split & Stitch functionality. This feature divides long media input files into shorter pieces that are transcoded in parallel and the results are joined together in the output file. In this way you can transcode large jobs a lot faster than real time. Split & Stitch operates transparently, with the files designated for splitting-and-stitching freely mixed with conventionally transcoded material, making maximally efficient use of the CPU power in your cluster.

Speed
Split-and-Stitch (turbo encoding)
Parallel job processing
Multi-core architecture (maximum utilization of processors)

High Availability
Automatic re-submission in case of a slave failure
Redundant master assures no single point of failure

High Availability

Input Monitoring

with support for: FTP, SMB, Avid Systems and Local Files

Dragonflies.mov

Firemagic.mov

Sunburst.mov

Waterspell.mov

Episode Engine Controller Node

Episode Engine Encoding Nodes

Maximal usage

Watch Folder

Event Action Scripts for cross platform integration

FTP

DAM

CMS

Proxy

Mobile

Web

IPTV

Automation   **Speed**   **High Availability**   **Integration**

Best practices – Usage

In order to maximize the experience and encoding speed of Episode Engine, different options for designing the optimal workflow according to your specific needs are described below.

File handling

For all Episode Engine nodes in the compression cluster to work at full speed, make sure the nodes have access to all input files as soon as possible, rather than adding files once the prior jobs are completed. This allows Episode Engine to divide the workload in an optimal way between all nodes.

For very large input files it is recommended to use symbolic linking instead of copying the file. By using symbolic linking you save not only time in transfer but also storage space.

For further optimization of your workflow use the ability to set different priority on the watchfolders to make sure the most urgent jobs get processed first.

Monitoring

Engine Admin is an application included with your Episode Engine, and it is used for setting up input monitors and monitoring the nodes of the cluster. It monitors both the status of the nodes, as well as the jobs that are being encoded. As an administrator, Engine Admin allows you to stop jobs that are being run, delete jobs that are queued, and to change the order of jobs in the queue.

Another available option for monitoring your workflow is to connect Episode Engine to an existing monitoring system through SNMP. With this you can always monitor your encoding nodes in the same system as all your other systems and ensure the appropriate action is taken.

Integration through Event actions

Episode Engine is normally deployed to work together with other systems for creating an entire automated workflow. The integration between Episode Engine and surrounding systems is handled through event actions. Once a transcoded job is finished, a set of user-defined scripts is invoked to make sure that the proper actions are taken. The possible actions are unlimited and common tasks include, for example, moving the output to a new location or notifying a content management system.

The system of event actions is completely configurable and any application that is executable on the system can be used.

In addition, event actions allow for defining a proper action to be taken if a job is not successful for some reason. This part is also fully configurable by the user in the same way as finished jobs are.

Flip4Mac®
Digital Media Tools for the Mac

TELESTREAM